WHAT IS CLAIMED IS:

1. A method for detecting units present on a network bus, the method comprising:

(a) monitoring a token rotation to determine the units which are present on the network bus;

(b) building an initial map of the network bus according to the monitored token rotation;

(c) monitoring a subsequent token rotation to determine the units which are present on the network bus at the subsequent time;

(d) building a new map of the network bus according to the subsequent token rotation by storing the addresses which respond to the transmitted token rotation;

(e) comparing the initial and new maps of the network bus to determine any differences therebetween;

(f) sending a free buffer enquiry to each address which is determined to be missing from the initial map; and

(g) building a master map of the units present on the network bus based on the new map and the responses to the free buffer enquiries.

2. The method according to claim 1, further comprising the step of updating and reinitializing the initial and new maps if the comparing step determines that the initial and new maps are different.

3. The method according to claim 1, further comprising the step of writing a one into memory to indicate a response to the transmitted token rotation and writing a zero into memory to indicate no response has been received from the transmitted token rotation.

4. The method according to claim 1, further comprising the step of writing a one into memory to indicate a response to the free buffer enquiry and writing a zero into memory to indicate no response has been received from the free buffer enquiry.

5. The method according to claim 3, wherein the sending of the free buffer enquiry is performed by searching the initial map from the lowest to the highest value for ones.

6. The method according to claim 3, wherein the sending of the free buffer enquiry is performed by searching the initial map for the next address with a one as long as a one exists in the old map.

7. The method according to claim 1, further comprising the step of tolerating a predetermined number of bit errors on the responses to the free buffer enquiries.

8. The method according to claim 1, further comprising repeating steps (a)-(g) wherein the new map from the previous token rotation becomes the initial map.

9. An apparatus for detecting units present on a network bus, the apparatus comprising:

means for monitoring a token rotation to determine the units which are present on the network bus;

means for building an initial map of the network bus according to the monitored token rotation;

means for monitoring a subsequent token rotation to determine the units which are present on the network bus at the subsequent time;

means for building a new map of the network bus according to the subsequent token rotation by storing the addresses which respond to the transmitted token rotation;

means for comparing the initial and new maps of the network bus to determine any differences therebetween;

means for sending a free buffer enquiry to each address which is determined to be missing from the initial map; and

means for building a master map of the units present on the network bus based on the new map and the responses to the free buffer enquiries.

10. The apparatus according to claim 9, further comprising means for updating and reinitializing the initial and new maps if the means for comparing determines that the initial and new maps are different.

11. The apparatus according to claim 9, further comprising means for writing a one into memory to indicate a response to the transmitted token rotation and writing a zero into memory to indicate no response has been received from the transmitted token rotation.

12. The apparatus according to claim 9, further comprising means for writing a one into memory to indicate a response to the free buffer enquiry and writing a zero into memory to indicate no response has been received from the free buffer enquiry.

13. The apparatus according to claim 11, wherein the sending of the free buffer enquiry is performed by searching the initial map from the lowest to the highest value for ones.

14. The apparatus according to claim 11, wherein the sending of the free buffer enquiry is performed by searching the initial map for the next address with a one as long as a one exists in the old map.

15. The apparatus according to claim 9, further comprising means for tolerating a predetermined number of bit errors on the responses to the free buffer enquiries.

16. The apparatus of claim 9, further comprising a snooper device located between a token bus and a bus processing device such that the snooper device can gain control of the token by intercepting tokens destined for the bus processing device, thereby not requiring the snooper device to have its own token bus address.

17. The apparatus of claim 9, further comprising a snooper device located in parallel with a bus processing device, wherein the snooper device has its own token bus address such that the snooper device can gain control of the token bus in the same manner as other units on the bus.

18. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for detecting units present on a network bus, the method comprising:

monitoring a token rotation to determine the units which are present on the network bus;

building an initial map of the network bus according to the monitored token rotation;

monitoring a subsequent token rotation to determine the units which are present on the network bus at the subsequent time;

building a new map of the network bus according to the subsequent token rotation by storing the addresses which respond to the transmitted token rotation;

comparing the initial and new maps of the network bus to determine any differences therebetween;

sending a free buffer enquiry to each address which is determined to be missing from the initial map; and

building a master map of the units present on the network bus based on the new map and the responses to the free buffer enquiries.

19. The program storage device according to claim 18, further comprising the step of updating and reinitializing the initial and new maps if the comparing step determines that the initial and new maps are different.

20. The program storage device according to claim 18, further comprising the step of writing a one into memory to indicate a response to the transmitted token rotation and writing a zero into memory to indicate no response has been received from the transmitted token rotation.

21. The program storage device according to claim 18, further comprising the step of writing a one into memory to indicate a response to the free buffer enquiry and writing a zero into memory to indicate no response has been received from the free buffer enquiry.

22. The program storage device according to claim 20, wherein the sending of the free buffer enquiry is performed by searching the initial map from the lowest to the highest value for ones.

23. The program storage device according to claim 20, wherein the sending of the free buffer enquiry is performed by searching the initial map for the next address with a one as long as a one exists in the old map.

24. A computer program product embodied in a computer-readable medium for detecting units present on a network bus, the computer program product comprising:

computer readable program code means for monitoring a token rotation to determine the units which are present on the network bus;

computer readable program code means for building an initial map of the network bus according to the monitored token rotation;

computer readable program code means for monitoring a subsequent token rotation to determine the units which are present on the network bus at the subsequent time;

computer readable program code means for building a new map of the network bus according to the subsequent token rotation by storing the addresses which respond to the transmitted token rotation;

computer readable program code means for comparing the initial and new maps of the network bus to determine any differences therebetween;

computer readable program code means for sending a free buffer enquiry to each address which is determined to be missing from the initial map; and;

computer readable program code means for building a master map of the units present on the network bus based on the new map and the responses to the free buffer enquiries.

25. The computer program product according to claim 24, further comprising computer readable program code means for updating and reinitializing the initial and new maps if the comparing step determines that the initial and new maps are different.

26. The computer program product according to claim 24, further comprising computer readable program code means for writing a one into memory to indicate a response to the transmitted token rotation and writing a zero into memory to indicate no response has been received from the transmitted token rotation.

27. The computer program product according to claim 24, further comprising computer readable program code means for writing a one into memory to indicate a response to the free buffer enquiry and writing a zero into memory to indicate no response has been received from the free buffer enquiry.

28. The computer program product according to claim 26, wherein the sending of the free buffer enquiry is performed by searching the initial map from the lowest to the highest value for ones.

29. The computer program product according to claim 26, wherein the sending of the free buffer enquiry is performed by searching the initial map for the next address with a one as long as a one exists in the old map.